PYTHON
Day00

Table of contents

ex00 - Hello World

filename: 00_hello_world.py

Instruction :
Display “Hello World” in the command line.

```
$ python 00_hello_world.py
Hello World
```

Hint if you're stuck 😉

ex01 - Print argument

filename: 01_print_arg.py

Instruction :
Display the first argument in the command line.

```
$ python 01_print_arg.py "this is the first argument"
this is the first argument
```

Hint if you're stuck 😉

ex02 - Reverse

filename: 02_reverse.py

Instruction :
Reverse the order of a string and swap its case.

```
$ python 02_reverse.py "Hello World"
DLROw OLLEh
$ python 02_reverse.py "Hello" "World"
DLROw OLLEh
$ python 02_reverse.py "Hello" World test
TSET DLROw OLLEh
```

Hint if you're stuck (1) 😉
Hint if you're stuck (2) 😉

ex03 - Odd or Even

filename: 03_odd_or_even.py

Instruction :
Check if a number is odd, even or zero.

```
$ python 03_odd_or_even.py 1
I'm Odd.
$ python 03_odd_or_even.py "2"
I'm Even.
$ python 03_odd_or_even.py 0
I'm Zero.
$ python 03_odd_or_even.py azerty
ERROR
$ python 03_odd_or_even.py 1 3
ERROR
```

Hint if you're stuck (1) 😉
Hint if you're stuck (2) 😉

ex04 - Text analyzer

filename: 04_text_analyzer.py

Instruction :
Create a function called text_analyzer that displays the sums of upper characters, lower characters, punctuation characters and spaces in a given text.

```
$ python 04_text_analyzer.py "Hello world! How are you today?"
The text contains 31 characters:

- 2 upper letters

- 22 lower letters

- 2 punctuation marks

- 5 spaces
```

Hint if you're stuck (1) 😉
Hint if you're stuck (2) 😉

ex05 - Operations

filename: 05_operations.py

Instruction :
Print the results of the addition, subtraction, multiplication, division and the modulo operation of two numbers.

```
$ python 05_operations.py 15 6
Sum:            21
Difference:     9
Product:        90
Quotient:       2.5
Remainder:      3
$ python 05_operations.py 15 0
Sum:            15
Difference:     15
Product:        0
Quotient:       ERROR (div by zero)
Remainder:      ERROR (modulo by zero)
$ python 05_operations.py
Usage: python 05_operations.py <number1> <number2>
Example:
        python 05_operations.py 10 3
$ python 05_operations.py 15 6 8
Input Error: too many arguments
Usage: python 05_operations.py <number1> <number2>
Example:
        python 05_operations.py 10 3
$ python 05_operations.py 15 "test"
Input Error: only numbers
Usage: python 05_operations.py <number1> <number2>
Example:
        python 05_operations.py 10 3
```

ex06 - Recipe

filename: 06_recipe.py

Instruction :
First, you will have to create a cookbook **dictionary** called *cookbook*.
cookbook will store 3 recipes :

- *sandwich*
- *cake*
- *salad*

Each recipe will store 3 values :

- *ingredients* : a list of ingredients
- *meal* : type of meal
- *prep_time* : preparation time in minutes

Sandwich's ingredients are ham, bread, cheese and tomatoes. It is a lunch and it takes 10 minutes of preparation.
Cake's ingredients are flour, sugar and eggs. It is a dessert and it takes 60 minutes of preparation.
Salad's ingredients are avocado, arugula, tomatoes and spinach. It is a lunch and it takes 15 minutes of preparation.

1. Get to know **dictionaries**. In the first place, try to print only the keys of the **dictionary**. Then only the values. And to conclude, all the items.
2. Write a function to print a recipe from *cookbook*. The function parameter will be : name of the recipe.
3. Write a function to delete a recipe from the dictionary. The function parameter will be: name of the recipe.
4. Write a function to add a new recipe to *cookbook* with its ingredients, its meal type and its preparation time. The function parameters will be: name of recipe, ingredients, meal and prep_time.
5. Write a function to print all recipe names from *cookbook*. Think about formatting the output.
6. Last but not least, make a program using the four functions you just created. The program will prompt the user to make a choice between printing the cookbook, printing only one recipe, adding a recipe, deleting a recipe or quitting the cookbook.

```
def printRecipe(recipe_name):
    pass
def deleteRecipe(recipe_name):
    pass
def addRecipe(recipe_name, ingredients, meal, prep_time):
    pass
def printAllRecipeNames():
    pass
```

```
$ python 06_recipe.py
Please select an option by typing the corresponding number:
1: Add a recipe
2: Delete a recipe
3: Print a recipe
4: Print the cookbook
5: Quit
>> 4
Recipes in the cookbook:

['sandwich', 'cake', 'salad']

Please select an option by typing the corresponding number:
1: Add a recipe
2: Delete a recipe
3: Print a recipe
4: Print the cookbook
5: Quit
>> 1
Please enter the recipe's name:
>> pasta
Please enter the recipe's ingredients separated by commas:
>> pasta,pesto
Please enter the recipe's meal type:
>> lunch
Please enter the recipe's preparation time:
>> 7

Please select an option by typing the corresponding number:
1: Add a recipe
2: Delete a recipe
3: Print a recipe
4: Print the cookbook
5: Quit
>> 2
Please enter the recipe's name to delete it:
>> cake
Recipe deleted!
Please select an option by typing the corresponding number:
1: Add a recipe
2: Delete a recipe
3: Print a recipe
4: Print the cookbook
5: Quit
>> test
This option does not exist, please type the corresponding number.

Please select an option by typing the corresponding number:
1: Add a recipe
2: Delete a recipe
3: Print a recipe
4: Print the cookbook
5: Quit
>> 3
Please enter the recipe's name to get its details:
```

```
>> salad
Recipe for salad:
Ingredients list: ['avocado', 'arugula', 'tomatoes', 'spinach']
To be eaten for lunch.
Takes 15 minutes of cooking.

Please select an option by typing the corresponding number:
1: Add a recipe
2: Delete a recipe
3: Print a recipe
4: Print the cookbook
5: Quit
>> 5
Cookbook closed.
```

Hint if you're stuck (1) 😉
Hint if you're stuck (2) 😉

ex07 - Words filter

filename: 07_filterwords.py

Instruction :
Make a program that removes all the words in a string that are shorter than or equal to n letters. You have to use **list comprehensions**.

```
$ python 07_filterwords.py
ERROR
$ python 07_filterwords.py "Hello my dear world"
ERROR
$ python 07_filterwords.py "Hello my dear world" 3
['Hello', 'dear', 'world']
$ python 07_filterwords.py "Hello my dear world" 4
['Hello', 'world']
```

Hint if you're stuck 😉

ex08 - SOS

filename: 08_sos.py

Instruction :
Make a program that removes all the words in a string that are shorter than or equal to n letters. You have to use **list comprehensions**. You can check morse code here.

```
$ python 08_sos.py "SOS"
... --- ...
$ python 08_sos.py "Hello" World
.... . .-.. .-.. ---  /  .-- --- .-. .-.. -..
$ python 08_sos.py "Hello / World"
ERROR
```

Hint if you're stuck 😉

ex09 - Secret number

filename: 09_who_am_i.py

Instruction :
Make a program that defines a random number between 1 and 99 and that asks the user to guess it. It will tell if the input is too low or too high and count the user's attempts. The game ends when the user finds the number or type ‘exit’.

```
$ python 09_who_am_i.py
This is an interactive guessing game!
You have to enter a number between 1 and 99 to find out the secret number. Type
'exit' to end the game.
Good luck!

What's your guess between 1 and 99?
>> 50
Too low!
What's your guess between 1 and 99?
>> 75
Too high!
What's your guess between 1 and 99?
>> 62
Congratulations, you've got it!
You won in 3 attempts!
$ python 09_who_am_i.py
[...]
What's your guess between 1 and 99?
>> 25
Congratulations, you got it on your first try!
$ python 09_who_am_i.py
[...]
What's your guess between 1 and 99?
>> 50
Too high!
What's your guess between 1 and 99?
>> 42
The answer to the ultimate question of life, the universe and everything is 42.
Congratulations, you've got it!
You won in 2 attempts!
```

Hint if you're stuck (1) 😉
Hint if you're stuck (2) 😉